Style-aware Mid-level Representation for Discovering Visual Connections in Space and Time

Yong Jae Lee[*], Alexei A. Efros[*], and Martial Hebert
Robotics Institute, Carnegie Mellon University
{yjlee22,efros,hebert}@cs.cmu.edu

Abstract

We present a weakly-supervised visual data mining approach that discovers connections between recurring mid-level visual elements in historic (temporal) and geographic (spatial) image collections, and attempts to capture the underlying visual style. *In contrast to existing discovery methods that mine for patterns that remain visually consistent throughout the dataset, our goal is to discover visual elements whose appearance changes due to change in time or location; i.e., exhibit consistent stylistic variations across the label space (date or geo-location). To discover these elements, we first identify groups of patches that are style-sensitive. We then incrementally build correspondences to find the same element across the entire dataset. Finally, we train style-aware regressors that model each element's range of stylistic differences. We apply our approach to date and geo-location prediction and show substantial improvement over several baselines that do not model visual style. We also demonstrate the method's effectiveness on the related task of fine-grained classification.*

1. Introduction

"Learn how to see. Realize that everything connects to everything else." – Leonardo da Vinci

Long before the age of "data mining", historians, geographers, anthropologists, and paleontologists have been discovering and analyzing patterns in data. One of their main motivations is finding patterns that correlate with spatial (geographical) and/or temporal (historical) information, allowing them to address two crucial questions: *where?* (geolocalization) and *when?* (historical dating). Interestingly, many such patterns, be it the shape of the handle on an Etruscan vase or the pattern of bark of a Norwegian pine, are predominantly *visual*. The recent explosion in the sheer volume of visual information that humanity has been capturing poses both a challenge (it's impossible to go through by hand), and an opportunity (discovering things that would never have been noticed before) for these fields. In this work, we take the first steps in considering temporally- as well as spatially-varying visual data and developing a method for *automatically discovering* visual patterns that correlate with time and space.

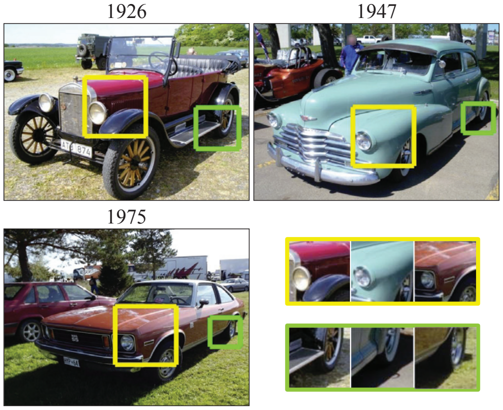


Figure 1. Given historic car images, our algorithm is not only able to automatically discover corresponding visual elements (e.g., yellow, green boxes) despite the large visual variations, but can model these variations to capture the changes in visual style across time.

Of course, finding recurring visual patterns in data underlies much of modern computer vision itself – it is what connects the disparate fragments of our visual world into a coherent narrative. At the low level, this is typically done via simple unsupervised clustering (e.g., k-means in visual words [25]). But clustering visual patterns that are more complex than simple blobs, corners and oriented bars turns out to be rather difficult because everything becomes more dissimilar in higher dimensions. The emerging subfield of visual category discovery/visual data mining [24, 9, 14, 19, 5, 23, 4, 21, 6] proposes ways to address this issue. Most such approaches look for tight clumps in the data, discovering visual patterns that stay globally consistent throughout the dataset. More recent discriminative methods, such as [23, 4], take advantage of weak supervision to divide the dataset into discrete subsets (e.g., kitchen vs. bathroom [23], Paris vs. Not-Paris [4]) to discover specific visual patterns that repeatedly occur in one subset while *not* occurring in others.

[*] Now with the EECS department at UC Berkeley.

But in addition to the globally-consistent visual patterns (e.g., the Pepsi logo is exactly the same all over the world) and the specific ones (e.g., toilets are only found in bathrooms), much in our visual world is neither global nor specific, but rather undergoes a *gradual visual change*. This is nowhere more evident than in the visual changes across large extents of space (geography) and time (history). Consider the three cars shown in Figure 1: one antique, one classic, and one from the 1970s. Although these cars are quite different visually, they clearly share some common elements, e.g., a headlight or a wheel. But notice that even these "common" elements differ substantially in their appearance across the three car types, making this a very challenging correspondence problem. Notice further that the way in which they differ is not merely random (i.e., a statistical "noise term"). Rather, these subtle yet consistent differences (curvy vs. boxy hood, the length of the ledge under the door, etc.) tend to reflect the particular *visual style* that is both specific to an era yet changing gradually over time (Figure 8). If now we were given a photo of a different car and asked to estimate its model year, we would not only need to detect the common visual elements on the new car but also understand what its stylistic differences (e.g., the length of that ledge) tell us about its age.

In this paper, we propose a method for discovering connections between similar mid-level visual elements in temporally- and spatially-varying datasets and modeling their "visual style". Here we define visual style as appearance variations of the same visual element due to change in time or location. Our central idea is to 1) create reliable *generic* visual element detectors that "fire" across the entire dataset independent of style, and then 2) model their *style-specific* differences using weakly-supervised image labels (date, geo-location, etc.). The reason for doing the first step is that each generic detector puts all of its detections into correspondence (lower right in Figure 1), creating a "closed-world" focused on one visual theme, where it is much easier to "subtract away" the commonalities and focus on the stylistic differences. Furthermore, without conditioning on the generic detector, it would be very difficult to even detect the stylistically-informative features. For instance, the ledge in Figure 1 (green box) is so tiny that it is unlikely to be detectable in isolation, but in combination with the wheel and part of the door (the generic part), it becomes highly discriminable.

We evaluate our method on the task of date and geo-location prediction in three scenarios: two historic car datasets with model year annotations and a Street View imagery dataset annotated with GPS coordinates. We show that our method outperforms several baselines, which do not explicitly model visual style. Moreover, we also demonstrate how our approach can be applied to the related task of fine-grained recognition of birds.

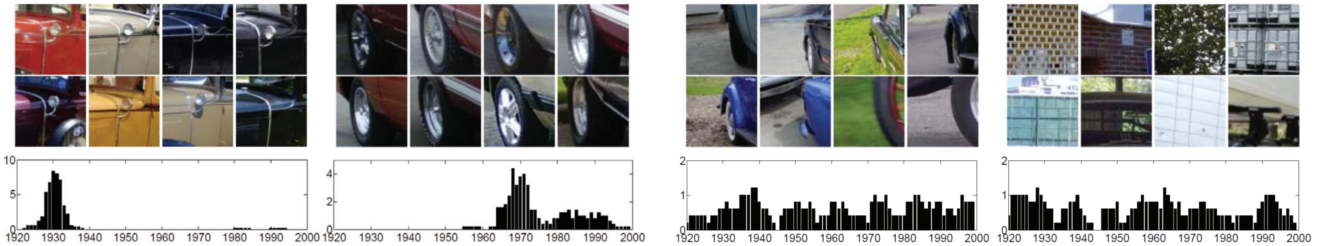
2. Related work

Modeling Space and Time Geo-tagged datasets have been used for geo-localization on the local [22, 12], regional [2], and planetary [10] scales, but we are not aware of any prior work on improving geo-location by explicitly capturing stylistic differences between geo-informative visual elements (but see [4] for anecdotal evidence of such possibility). Longitudinal (i.e., long-term temporal) visual modeling has received relatively little attention. Most previous research has been on the special case of age estimation for faces (see [8] for a survey). Recent work includes modeling the temporal evolution of Web image collections [11] and dating of historical color photographs [17]. We are not aware of any prior work on modeling historical visual style.

Visual data mining Existing visual data mining/object discovery approaches have been used to discover object categories [24, 9, 19, 6], mid-level patches [23, 4, 20], attributes [5, 21], and low-level foreground features [14]. Typically, an appropriate similarity measure is defined between visual patterns (i.e., images, patches, or contours) and those that are most similar are grouped into discovered entities. Of these methods, mid-level discriminative patch mining [23, 4] shares the most algorithmic similarities with our work; we also represent our visual elements with HOG patches [3] and refine the clusters through discriminative cross-validation training. However, unlike [23, 4] and all existing discovery methods, we go beyond simply detecting recurring visual elements, and model the stylistic differences among the common discovered elements.

Visual style analysis The seminal paper on "style-content separation" [26] uses bilinear models to factor out the style and content components in pre-segmented, pre-aligned visual data (e.g., images of letters in different fonts). While we also use the term "style" to describe the differences between corresponding visual elements, we are solving a rather different problem. Our aim is to automatically *discover* recurring visual elements despite their differences in visual style, and then model those differences. While our "generic detectors" could perhaps be thought of as capturing "content" (independent of style), we do not explicitly factor out the style, but model it *conditioned* on the content.

Fine-grained categorization can also be viewed as a form of style analysis, as subtle differences within the same basic-level category differentiate one subordinate category from another. Existing approaches use human-labeled attributes and keypoint annotations [28, 7, 32, 1] or template matching [31, 30]. Because these methods are focused on classification, they limit themselves to the simpler visual world of manually-annotated object bounding boxes, whereas our method operates on full images. Furthermore, discovering one-to-one correspondences is given a primary role in our method, whereas in most fine-grained ap-



(a) Peaky (low-entropy) clusters

(b) Uniform (high-entropy) clusters

Figure 2. Mining style-sensitive visual elements. Clusters are considered style-sensitive if they have "peaky" (low-entropy) distribution across time (a) and style-insensitive if their instances are distributed more uniformly (b). Notice how the high-entropy distributions (b) represent not only style insensitivity (e.g., nondescript side of car) but also visually-noisy clusters. Both are disregarded by our method.

proaches the correspondences are already provided. While template matching methods [31, 30] also try to discover correspondences, unlike our approach, they do not explicitly model the style-specific differences within each correspondence set. Finally, these approaches have not been applied to problems with continuous labels (regression), where capturing the range of styles is particularly important.

Lastly, relative attributes [18] model how objects/scenes relate to one another via ordered pairs of labels (A is "furrier" than B). We also share the idea of relating things. However, instead of using strong supervision to define these relationships, we automatically mine for visual patterns that exhibit such behavior.

3. Approach

Our goal is to discover and connect mid-level visual elements across temporally- and spatially-varying image collections and model their style-specific differences. We assume that the image collections are weakly supervised with date or location labels.

There are three main steps to our approach: First, as initialization, we mine for "style-sensitive" image patch clusters, that is, groups of visually similar patches with similar labels (date or location). Then, for each initial cluster, we try to generalize it by training a generic detector that computes correspondences across the entire image collection to find the same visual element independent of style. Finally, for each set of correspondences, we train a style-aware regression model that learns to differentiate the subtle stylistic differences between different instances of the same generic element. In the following sections, we describe each of the steps in turn. We will use an image collection of historic cars as our running example, but note that there is nothing specific to cars in our algorithm.

3.1. Mining style-sensitive visual elements

Most recurring visual patterns in our data will be extremely boring (sky, asphalt, etc.). They will also not exhibit any stylistic variation over time (or space), and not be of any use in historical dating (or geo-localization) – after all, asphalt is always just asphalt! Even some parts of the car (e.g., a window) do not really change much over the decades. On the other hand, we would expect the shape of the hood between two 1920s cars to be more similar than between a 1920s and a 1950s car. Therefore, our first task is to mine for visual elements whose appearance somehow correlates with its labels (i.e., date or location). We call visual elements that exhibit this behavior *style-sensitive*.

Since we do not know *a priori* the correct scale, location, and spatial extent of the style-sensitive elements, we randomly sample patches across various scales and locations from each image in the dataset. Following [4], we represent each patch with a Histogram of Gradients (HOG) descriptor [3], and find its top N nearest neighbor patches in the database (using normalized correlation) by matching it to each image in a sliding window fashion over multiple scales and locations. To ensure that redundant overlapping patches are not chosen more than once, for each matching image we only take its best matching patch.

Each sampled patch and its N nearest neighbors ideally form a cluster of a recurring visual element (although many clusters will be very noisy due to inadequacies of simple HOG matching). To identify the style-sensitive clusters, we can analyze the temporal distribution of labels for each cluster's instances. Intuitively, a cluster that has a tightly-grouped ("peaky") label distribution suggests a visual element that prefers a particular time period, and is thus style-sensitive, while a cluster that has a uniform label distribution suggests a pattern that doesn't change over time. As extra bonus, most noisy clusters will also have a uniform distribution since it is very unlikely to be style-sensitive by random chance. To measure the style-sensitivity of cluster c, we histogram its labels and compute its entropy: $E(c) = -\sum_{i=1}^{n} H(i) \cdot \log_2 H(i)$, where $H(i)$ denotes the histogram count for bin i and n denotes the number of quantized label bins (we normalize the histogram to sum to 1). We then sort the clusters in ascending order of entropy. Figure 2 (a) and (b) show examples of the highest and lowest ranked clusters for the car dataset images. Notice how the highest ranked clusters correspond to style-sensitive car elements, while the lowest ranked clusters contain noisy or style-insensitive ones. We take the top M clusters as

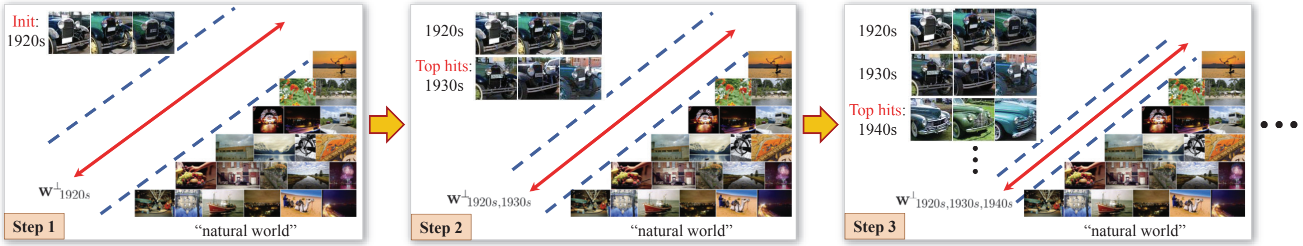
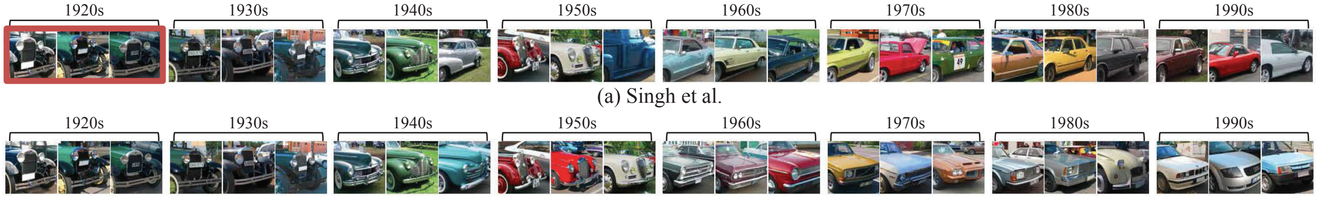


Figure 3. To account for a visual element’s variation in style over space or time, we incrementally revise its detector by augmenting the positive training set with the top detections fired only on images with “nearby” labels. This produces an accurate generic detector that is invariant to the visual element’s changes in style.



(a) Singh et al.

(b) Our approach

Figure 4. Establishing correspondences across time. (a) Correspondences made using the discriminative patch mining approach [23, 4] using a positive set of 1920s frontal cars. Note how the correspondences break down a third of the way through. (b) Starting with the same initial set of 1920s frontal cars, our algorithm gradually expands the positive set over the continuous label space until it is able to connect the same visual element across the entire temporal extent of the dataset.

our discovered style-sensitive visual elements, after rejecting near-duplicate clusters (measured by spatial overlap of more than 25% between any 5 of their cluster members).

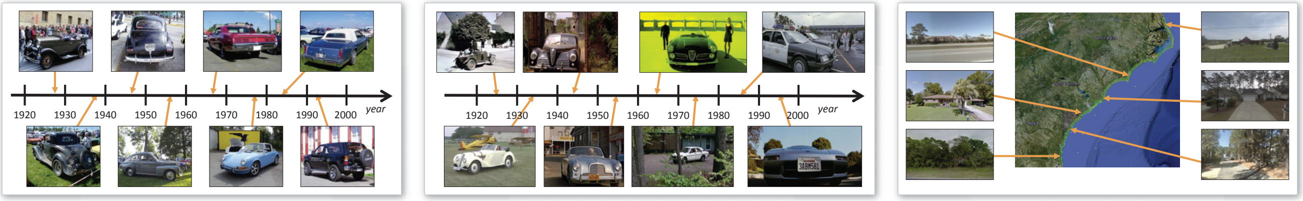
3.2. Establishing correspondences

Each of the top M clusters corresponds to a style-sensitive visual element in a local region of the label space. A few of these elements represent very specific visual features that just do not occur in other parts of the data (e.g., car tailfins from 1960s). But most others have similar counterparts in other time-periods and our goal is to connect them together, which will allow us to model the change in style of the same visual element over the entire label space. For instance, one of the style-sensitive elements could represent frontal cars from 1920s. We want to find corresponding frontal car patches across all time periods.

The same visual element, however, can look quite different across the label space, especially over larger temporal extents (Figure 1). To obtain accurate correspondences across all style variations, we propose to train a discriminative detector using an iterative procedure that exploits the continuous nature of the label space. In general, we expect the appearance of a visual element to change *gradually* as a function of its label. Our key idea is to initialize the detector using a style-sensitive cluster as the initial positive training data, but then incrementally revise it by augmenting the positive set with detections fired only on images with “nearby” labels (e.g., decades), as shown in Figure 3.

Specifically, we first train a linear SVM detector with the cluster patches as positives and patches sampled from thousands of random Flickr images as negatives. These negatives will make the detector discriminative against generic patterns occurring in the “natural world” [23], which helps it to fire accurately on unseen images. We then incrementally revise the detector. At each step, we run the current detector on a new subset of the data that covers a slightly broader range in label space, and retrain it by augmenting the positive training set with the top detections. We repeat this process until all labels have been accounted for. Making these transitive connections produces a final generic detector that fires accurately across the entire label space, as shown in Figure 4 (b). Note that automatic discovery of transitive visual correspondences across a dataset is very much in the spirit of the Visual Memex [16] opening up several promising future directions for investigation.

There is an important issue that we must address to ensure that the detector is robust to noise. The initial cluster can contain irrelevant, outlier patches, since some of the top N nearest neighbors of the query patch could be bad matches. To prune out the noisy instances, at each step of the incremental revision of our detector, we apply cross-validation training [23, 4]. Specifically, we create multiple partitions of the training set and iteratively refine the current detector by: (1) training on one partition; (2) testing on another; (3) taking the resulting top detections as the new training instances; and (4) repeating steps 1-3 until convergence, i.e., the top detections do not change. Effectively, at each iteration, the detector learns to boost the common patterns shared across the top detections and down-weights their discrepancies without over-fitting, which leads to more



(a) Car Database (CarDb)

(b) Internet Movie Car Database (IMCDb)

(c) East coast Database (EDb)

Figure 5. Each CarDb and IMCDb image is labeled with the car's model year. Each EDb image is labeled with its GPS coordinate.

accurate detections in the next iteration.

Note that a direct application of [23, 4] will not work for our case of continuous, style-varying data because the variability can be too great. Figure 4 (a) shows detections made by a detector trained with [23, 4], using the same initial style-sensitive cluster of 1920s cars as positives. The detector produces accurate matches in nearby decades, but the correspondence breaks down across larger temporal extents because it fails to model the variation in style.

Finally, we fire each trained generic detector on all images and take the top detection per image (and with SVM score greater than -1) to obtain the final correspondences.

3.3. Training style-aware regression models

The result of the previous step is a set of generic mid-level detectors, each tuned to a particular visual element and able to produce a set of corresponding instances under many different styles. Now we are finally ready to model that variation in style. And because the correspondences are so good, we can now forget about the larger dataset and focus entirely on each set of corresponding instances in isolation, making our modeling problem much simpler. The final step is to train a style-aware regressor for each element that models its stylistic variation over the label space.

It is safe to assume that style will not change linearly over the label space (e.g., with cars, it is possible that stylistic elements from one decade could be reintroduced as "vintage" in a later decade). To account for this, we train a standard non-linear Support Vector Regressor (SVR) with an ϵ-insensitive loss function using ground-truth weakly-supervised image labels (e.g., date, geo-location) as the target score. We use Gaussian kernels: $K(x_i, x_j) = \exp(-\gamma^{-1}||x_i - x_j||^2)$, where γ is the mean of the pairwise distances among all instances and x_i is the HOG feature for instance i. Under this kernel, instances with similar appearance are most likely to have similar regression outputs. Furthermore, to handle possible mis-detections made by the generic detector which could add noise, we weight each instance proportional to its detection score when training the SVR. (We map a detection score s to a weight in [0,1], via a logistic function $1/(1 + \exp(-2s))$.) Each resulting model captures the stylistic differences of the same visual element found by the generic detector.

4. Results

In this section, we 1) evaluate our method's ability to predict date/location compared to several baselines, 2) provide in-depth comparisons to the discriminative patch mining approach of [23], 3) show qualitative examples of discovered correspondences and learned styles, and 4) apply our approach to fine-grained recognition of birds.

Datasets We use three datasets: 1) Car Database (CarDb): 13,473 photos of cars made in 1920-1999 crawled from www.cardatabase.net; 2) Internet Movie Car Database (IMCDb): 2,400 movie images of cars made in 1920-1999 crawled from www.imcdb.org; and 3) East Coast Database (EDb): 4,455 Google Street View images along the eastern coasts of Georgia, South Carolina, and North Carolina. Example images are shown in Figure 5. CarDb and IMCDb images are labeled with the model year of the main car in the image, and EDb images are labeled with their GPS coordinates. These are the "style" labels and the only supervisory information we use. For EDb, since our SVRs expect 1D outputs (although a multivariate regression method could also be used), we project the images' 2D GPS coordinates to 1D using PCA (this works because the area of interest is roughly linear, i.e., long and narrow, see Figure 5 (c)). These datasets exhibit a number of challenges including clutter, occlusion, scale, location and viewpoint change, and large appearance variations of the objects. Importantly, unlike standard object recognition datasets, ours have continuous labels. We partition the CarDb and EDb datasets into train/test sets with 70/30% splits. We evaluate on all datasets, and focus additional analysis on CarDb since it has the largest number of images.

Image-level date/location prediction To evaluate on a label prediction task, we need to combine all of our visual element predictors together. We train an image-level prediction model using as features the outputs of each style-aware regressor on an image. Specifically, we represent an image I with feature $\phi(I)$, which is the concatenation of the maximum SVM detection scores of the generic detectors (over the image) and the SVR scores of their corresponding style-aware regressors. When testing on EDb, we aggregate the features in spatial bins via a spatial pyramid [13, 15], since we expect there to be spatial consistency of visual patterns across images. We use these features to train an image-level

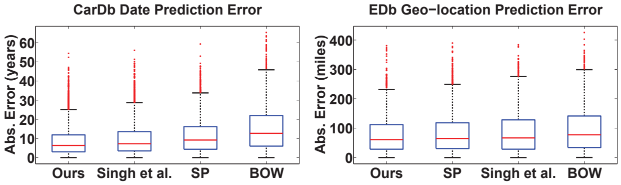


Figure 6. Box plots showing date and location prediction error on the CarDb and EDb datasets, respectively. Lower values are better. Our approach models the subtle stylistic differences for each discovered element in the data, which leads to lower error rates.

	Ours	Singh et al. [23, 4]	SP [13]	BOW
CarDb (years)	**8.56**	9.72	11.81	15.39
EDb (miles)	**77.66**	87.47	83.92	97.78
IMCDb (years)	**13.53**	15.32	17.06	18.65

Table 1. Mean absolute error on CarDb, EDb, and IMCDb for all methods. The result on IMCDb evaluates cross-dataset generalization performance. Lower values are better.

Gaussian SVR. This model essentially selects the most useful style-aware regressors for predicting style given the entire image. To ensure that the image-level model does not overfit, we train it on a separate validation set.

Baselines For date/location prediction, we compare to three baselines: bag-of-words (BOW), spatial pyramid (SP) [13], and Singh et al. [23]. For the first two, we detect dense SIFT features, compute a global visual word dictionary on the full dataset, and then train an intersection-kernel SVR using the date/location labels. For Singh et al. [23], which mines discriminative patches but does not model their change in style, we adapt the approach to train date/location-specific patch detectors using the initial style-sensitive clusters discovered in Sec. 3.1. Specifically, we take each specific cluster's instances as positives and all patches from the remaining training images that do not share the same labels (with a small "don't care" region in between) as negatives. Now, just like in the previous paragraph, we concatenate the max output of the detectors as features to train an image-level Gaussian SVR. We optimize all baselines' parameters by cross-validation.

Implementation details We sample 80x80 pixel patches over an image pyramid at 5 scales (i.e., min/max patch is 80/320 pixels wide in original image), and represent them with a 10x10x31 HOG descriptor [3]. For EDb patches, we augment HOG with a 10x10 tiny-image in Lab colorspace when training the style-aware SVRs. We set $N = 50$, $n = 80$, and $M = 80,\ 315$ for CarDb and EDb, respectively. For our generic SVM detectors, we fix $C_{svm} = 0.1$, and cover 1/8 of the label-space at each training step; CarDb: 10 years, EDb: 66 miles. For our SVRs, we fix $\epsilon = 0.1$ and set $C_{svr} = 100$ and 10 for CarDb and EDb, respectively, tuned using cross-validation on the training set.

4.1. Date and location prediction accuracy

We first evaluate our method's ability to predict the correct date/geo-location of the images in CarDb/EDb. Figure 6 and Table 1 (top rows) show the absolute error rates for all methods. This metric is computed by taking the absolute difference between the ground-truth and predicted labels.

Our approach outperforms all baselines on both datasets. The baselines have no mechanism to explicitly model stylistic differences as they are either mining discriminate patches over a subregion in label space (Singh et al.) or using quantized local features (BOW and SP) that result in loss of fine detail necessary to model subtle stylistic changes. Without explicitly making connections over space/time, the baselines appear to have difficulty telling apart signal from noise. In particular, we show substantial improvement on CarDb, because cars exhibit more pronounced stylistic differences across eras that require accurate modeling. The stylistic differences in architecture and vegetation for EDb are much more subtle. This makes sense, since the geographic region of interest only spans about 530 miles along the U.S. east coast. Still, our method is able to capture more of the stylistic differences to produce better results. Note that chance performance is around 19 years and 113 miles for CarDb and EDb, respectively; all methods significantly outperform chance, which shows that stylistic patterns correlated with time/location are indeed present in these datasets.

Figure 7 shows some discovered correspondences. Notice the stylistic variation of the car parts over the decades (e.g., windshield) and the change in amount/type of vegetation from north to south (e.g., trees surrounding the houses). In Figure 8 we visualize the learned styles of a few style-aware regressors on CarDb by averaging the most confident detected instances of each predicted decade.

4.2. Cross-dataset generalization accuracy

Recent work on dataset bias [27] demonstrated that training and testing on the same type of data can dramatically over-estimate the performance of an algorithm in a real-world scenario. Thus, we feel that a true test for an algorithm's performance should include training on one dataset while testing on a different one, whenever possible.

To evaluate cross-dataset generalization performance, we take the models trained on CarDb and test them on IMCDb. The third row in Table 1 shows the result. The error rates have increased for all methods compared to those on CarDb (with BOW now almost at chance level!). Overall, IMCDb is more difficult as it exhibits larger appearance variations due to more significant changes in scale, viewpoint, and position of the cars. CarDb, on the other hand, is a collection of photos taken by car-enthusiasts, and thus, the cars are typically centered in the image in one of a few canonical viewpoints. Note also that the gap between BOW and SP is smaller compared to that on CarDb. This

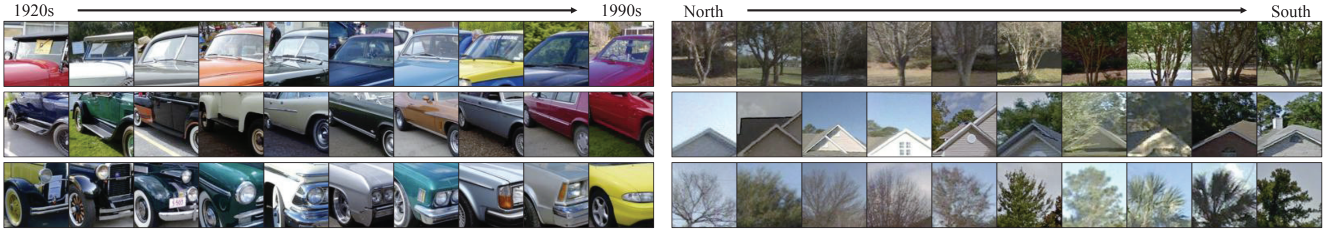


Figure 7. Example correspondences. Notice how a visual element's appearance can change due to change in time or location.

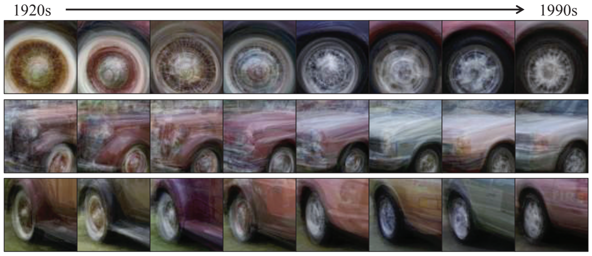


Figure 8. We visualize the styles that a single style-aware regressor has learned by averaging the predictions for each decade.

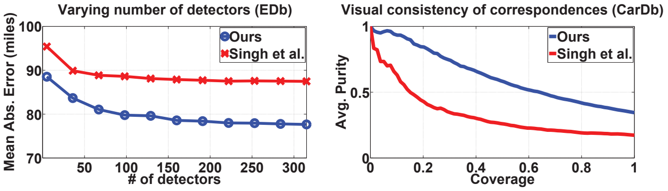


Figure 9. Varying the number of detectors (left), and visual consistency of correspondences (right). Lower/higher curves are better.

is mainly because spatial position is ignored in BOW while it is an important feature in SP. Since the objects' spatial position in IMCDb is more varied, SP tends to suffer from the different biases. Since our generic detectors are scale- and translation-invariant, we generalize better than the baselines. Singh et al. is also scale- and translation-invariant, and thus, shows better performance than BOW and SP. Still, ours retains a similar improvement over that baseline.

4.3. Detailed comparisons to Singh et al. [23]

In this section, we present detailed comparisons to Singh et al. [23], which is similar to our method but does not capture the style-specific differences.

Robustness to number of detectors Figure 9 (left) plots the geo-location prediction error as a function of the number of detectors on EDb for the two methods (the curve averages the error rates over five runs; in each run, we randomly sample a fixed number of detectors, and corresponding style-aware models for ours, among all 315 detectors to train the image-level SVR). Our approach outperforms the baseline across all points, saturating at a much lower error rate. This result demonstrates that when the visual patterns in the data change subtly, we gain a lot more from being style-aware than being discriminative.

We also analyze how well our models generalize across the label space. Using generic detectors initialized only with the visual patterns discovered within a specific decade (which results in 10 detectors), we train the corresponding style-aware regression models. We then use their outputs to train the image-level regressor. Across all eight different decade initializations, we find our final mean prediction error rates to be quite stable ($\sim$10 years). This shows our approach's generalizability and robustness to initialization.

Visual consistency of correspondences We next evaluate the quality of our discovered correspondences to that of Singh et al. using a purity/coverage plot. Purity is the % of cluster members that belong to the same visual element and coverage is the number of images covered by a given cluster. These are standard metrics used to evaluate discovery algorithms, and Singh et al. already showed superior performance over common feature/region clustering approaches using them. Thus, we feel it is important to evaluate our approach using the same metrics. We randomly sample 80 test images (10 per decade) from CarDb, and randomly sample 15 generic detectors and 15 discriminative detectors for ours and the baseline, respectively. We fire each detector and take its highest-scoring detected patch in each image. We sort the resulting set of detections in decreasing detection score, and ask a human labeler to mark the inliers/outliers (inliers are majority of high-scoring detections belonging to the same visual element). Using these human-marked annotations and treating each set of detections as a cluster, we compute average purity as a function of coverage.

Figure 9 (right) shows the result. We generate the curve by varying the threshold on the detection scores to define cluster membership and average the resulting purity scores (e.g., at coverage = 0.1, purity is computed using only the top 10% scoring detections in each cluster). Both ours and the baseline produce high purity when the clusters consist of only the highest-scoring detections. As more lower-scoring instances are included in the clusters, the baseline's purity rates fall quickly, while ours fall much more gracefully. This is because the baseline is trained to be discriminative against visual elements from other time periods. Thus, it succeeds in detecting corresponding visual elements that are consistent within the same period, but cannot generalize outside of that period well. Our detectors are trained to be generic and thus able to generalize much better, maintaining

	Ours	Zhang et al. [32]	Berg & Belhumeur [1]
Mean Accuracy (%)	41.01	28.18	56.89

Table 2. Fine-grained recognition on CUB-200-2011 [29].

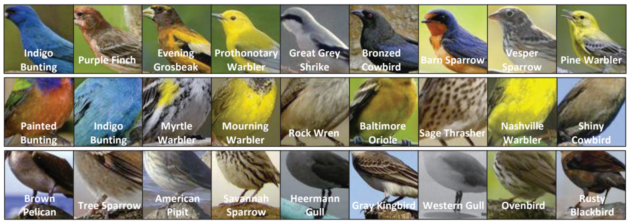


Figure 10. Example correspondences on CUB-200-2011.

high purity with increased coverage.

4.4. Fine-grained recognition

Finally, the idea of first making visual connections across a dataset to create a “closed world”, and then modeling the style-specific differences is applicable to several other domains. As one example, we adopt our method (with minor modifications) to the task of fine-grained recognition of bird species, where the labels are discrete.

Specifically, we first mine recurring visual elements that repeatedly fire inside the foreground bounding box (of any bird category) and not on the background (cf. style-sensitive clusters). We take the top-ranked clusters and train generic unsupervised bird-part detectors. Then, given the correspondence sets produced by each detector, we train 1-vs-all linear SVM classifiers to model the style-specific differences (cf. style-aware SVRs). Finally, we produce an image-level representation, pooling the maximum responses of the detectors and corresponding classifier outputs in a spatial pyramid. We use those features to train image-level 1-vs-all linear SVM classifiers.

We evaluate classification accuracy on the CUB-200-2011 dataset [29] using the provided bounding box annotations. We compare to the state-of-the-art methods of [32, 1], which define pose-normalized kernels and part-based features, respectively. To our knowledge, these are the only methods that test on all 200 categories on this dataset to date. Importantly, unlike our method, these baselines use ground-truth part annotations for training.

Table 2 shows mean classification accuracy over all 200 bird categories. While our approach performs worse than [1], it outperforms [32] despite using less supervision than either. We attribute this to our generic detectors producing accurate correspondences for the informative bird parts (see Figure 10), allowing our style-specific models to better discriminate the fine-grained differences.

Acknowledgements We thank Olivier Duchenne for helpful discussions. This work was supported in part by Google, ONR MURI N000141010934, and the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL or the U.S. Government.

References

[1] T. Berg and P. Belhumeur. POOF: Part-Based One-vs-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation. In *CVPR*, 2013. 2, 8

[2] M. Cristani, A. Perina, U. Castellani, and V. Murino. Geolocated Image Analysis using Latent Representations. In *CVPR*, 2008. 2

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005. 2, 3, 6

[4] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What Makes Paris Look like Paris? In *SIGGRAPH*, 2012. 1, 2, 3, 4, 5, 6

[5] K. Duan, D.Parikh, D.Crandall, and K.Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012. 1, 2

[6] A. Faktor and M. Irani. Clustering by Composition Unsupervised Discovery of Image Categories. In *ECCV*, 2012. 1, 2

[7] R. Farrell, O. Oza, N. Zhang, V. Morariu, T. Darrell, and L. Davis. Birdlets: Subordinate Categorization Using Volumetric Primitives and Pose-Normalized Appearance. In *ICCV*, 2011. 2

[8] Y. Fu, G.-D. Guo, and T. Huang. Age Synthesis and Estimation via Faces: A Survey. *TPAMI*, 2010. 2

[9] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006. 1, 2

[10] J. Hays and A. Efros. Im2gps: Estimating Geographic Information from a Single Image. In *CVPR*, 2008. 2

[11] G. Kim, E. Xing, and A. Torralba. Modeling and Analysis of Dynamic Behaviors of Web Image Collections. In *ECCV*, 2010. 2

[12] J. Knopp, J. Sivic, and T. Pajdla. Avoiding Confusing Features in Place Recognition. In *ECCV*, 2010. 2

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006. 5, 6

[14] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning From Partially Matching Images. *IJCV*, 85, 2009. 1, 2

[15] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei. Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification. In *NIPS*, 2010. 5

[16] T. Malisiewicz and A. Efros. Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In *NIPS*, 2009. 4

[17] F. Palermo, J. Hays, and A. A. Efros. Dating Historical Color Images. In *ECCV*, 2012. 2

[18] D. Parikh and K. Grauman. Relative Attributes. In *ICCV*, 2011. 3

[19] N. Payet and S. Todorovic. From a Set of Shapes to Object Discovery. In *ECCV*, 2010. 1, 2

[20] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012. 2

[21] M. Rastegariy, A. Farhadi, and D. Forsyth. Attribute Discovery via Predictable Discriminative Binary Codes. In *ECCV*, 2012. 1, 2

[22] G. Schindler, M. Brown, and R. Szeliski. Cityscale Location Recognition. In *CVPR*, 2007. 2

[23] S. Singh, A. Gupta, and A. A. Efros. Unsupervised Discovery of Mid-level Discriminative Patches. In *ECCV*, 2012. 1, 2, 4, 5, 6, 7

[24] J. Sivic, B.Russell, A.Efros, A.Zisserman, and W.Freeman. Discovering object categories in image collections. In *ICCV*, 2005. 1, 2

[25] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV*, 2003. 1

[26] J. Tenenbaum and W. Freeman. Separating Style and Content with Bilinear Models. *Neural Computation*, 12(6), 2000. 2

[27] A. Torralba and A. A. Efros. Unbiased Look at Dataset Bias. In *CVPR*, 2011. 6

[28] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition part localization with humans in the loop. In *ICCV*, 2011. 2

[29] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 8

[30] S. Yang, L. Bo, J. Wang, and L. Shapiro. Unsupervised Template Learning for Fine-Grained Object Recognition. In *NIPS*, 2012. 2, 3

[31] B. Yao, A. Khosla, and L. Fei-Fei. Combining Randomization and Discrimination for Fine-Grained Image Categorization. In *CVPR*, 2011. 2, 3

[32] N. Zhang, R. Farrell, and T. Darrell. Pose Pooling Kernels for Subcategory Recognition. In *CVPR*, 2012. 2, 8